TOPIC 5 (Development of criticality codes and methods) for oral presentation

COG – PUBLICLY AVAILABLE NOW TO CRITICALITY SAFETY PRACTITIONERS

Richard M. Buck, Dermott E. Cullen, David P. Heinrichs[1], Edward M. Lent,
Dale E. Nielsen, Jr, Kenneth E. Sale

Lawrence Livermore National Laboratory[2]
7000 East Avenue, P.O. Box 808, L-198, Livermore, California, USA

Abstract

COG is a modern, general-purpose, high fidelity, multi-particle transport code with a long history of use in criticality safety studies at the Lawrence Livermore National Laboratory. This code was released to the Radiation Safety Information Computational Center (RSICC) for distribution to the public for the first time in January 2006. This paper provides an overview of the code development history, a description of features and capabilities of interest to the criticality safety practitioner, and our plans in support of the next public RSICC release.

Code development history and philosophy

COG[3] was originally developed in the 1980s to simulate radiation detection and signal processing in support of underground nuclear testing as well as to serve users at the Lawrence Livermore National Laboratory (USA) as a general-purpose code for shielding and criticality calculations. The basic design philosophy[4] of the COG code developers has been to develop a user-friendly code that can accurately solve general particle (coupled) transport problems in complex three-dimensional geometry. Accuracy is obtained through implementation of state-of-the-art physics models and nuclear databases with no approximations to "speed up" execution times. Instead, the use of the world's most powerful supercomputers has historically alleviated any concern about slow execution times.

Public release and website

Today, inexpensive personal computers provide memory and execution speed sufficient for COG to be useful to the general criticality safety practitioner or university student.

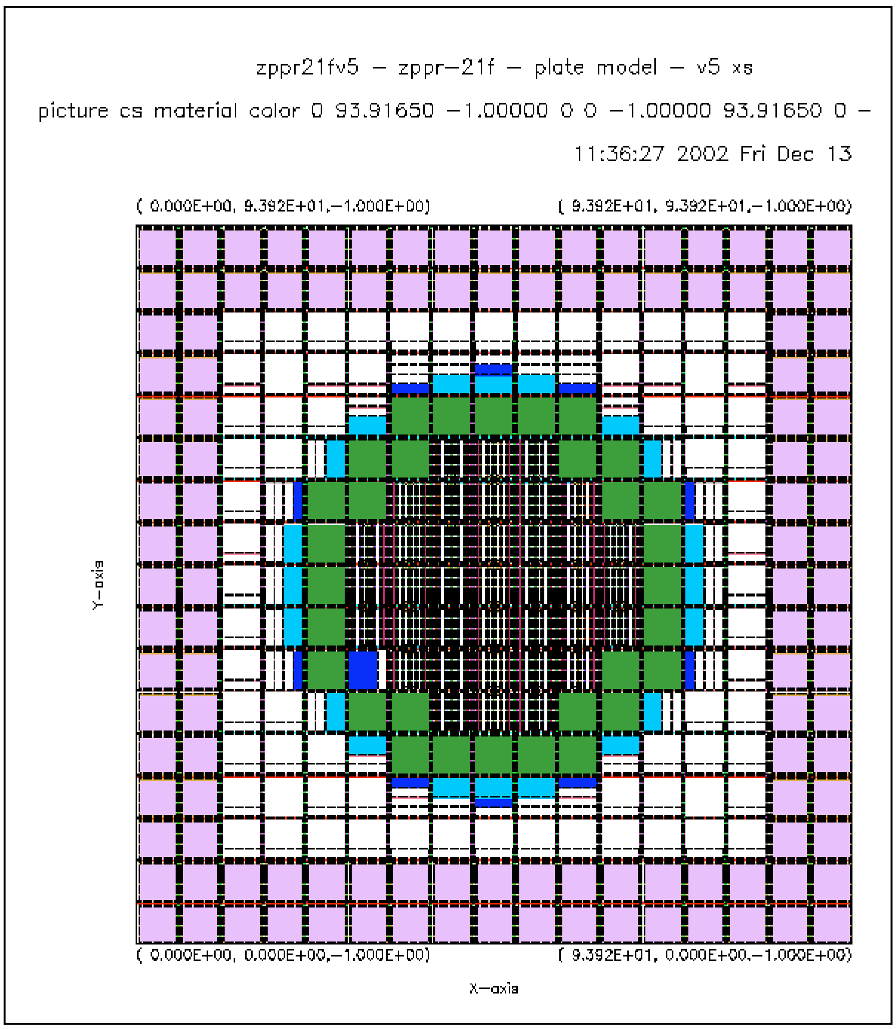
---

[1] heinrichs1@llnl.gov.
[2] This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.
[3] T. P. Wilcox, Jr., Nuclear Test-Experimental Science, Annual Report, FY 1989, UCRL-53929-89, page 97, *Charged-Particle Transport Features in the COG Code*.
[4] Richard Buck and Edward Lent, Energy & Technology Review, UCRL-52000-93-6, pages 9-16, *COG: A New, High-Resolution Code for Modeling Radiation Transport*.

Consequently, our first public release focused on the most commonly available personal computer and workstation platforms utilizing LINUX, SOLARIS or WINDOWS operating systems. New users are encouraged to register on our website[5] and download a copy of the User's Manual[6]. The website provides a link to order the code from RSICC and the email address[7] of the COG code development team. The website is also used as a means of distributing publications of tutorial value or general interest as well as advertising recent research activities to attract new collaborators.

Complex geometry capabilities

Three examples of complex geometries modeled using COG are presented below; namely, the ZPPR-21F assembly (Argonne National Laboratory), the 9975 Type B shipping container, and the Annular Test Reactor (Idaho National Laboratory).

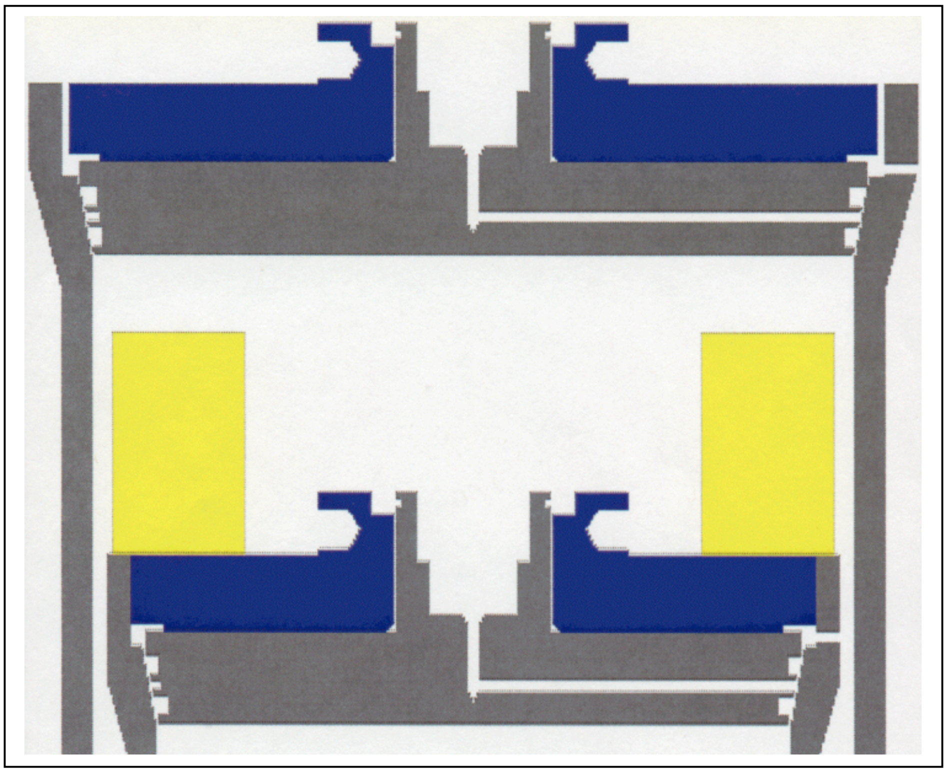


---

[5] http://cog.llnl.gov.
[6] Richard Buck, Edward Lent, Tom Wilcox and Stella Hadjimarkos, COG: A Multiparticle Monte Carlo Transport Code, User's Manual, Fifth Edition, UCRL-TM-202590, September 1, 2002.
[7] COG@llnl.gov.

Developing a model of a complex geometry system is often a time-consuming and difficult task. To make this task easier, the COG code developers provide closed surfaces such as BOX, HEXAHEDRON, PRISM and REVOLUTION[8] as well as complex surfaces such as TOPOGRAPHICAL[9].

Several (semi-automated) codes have been written to assist in translating complex geometry specifications (and other input) from other codes to COG. The figure above shows the COG model created from a VIM benchmark-model of the ZPPR-21F assembly as published in the ICSBEP Handbook using the LLNL-developed 'vim2cog' translator code. The VIM model consists of an assemblage of many plates defined by RPP (rectangular parallelepipedon) cards, which are easily machine translated into a corresponding assemblage of BOX cards in COG.

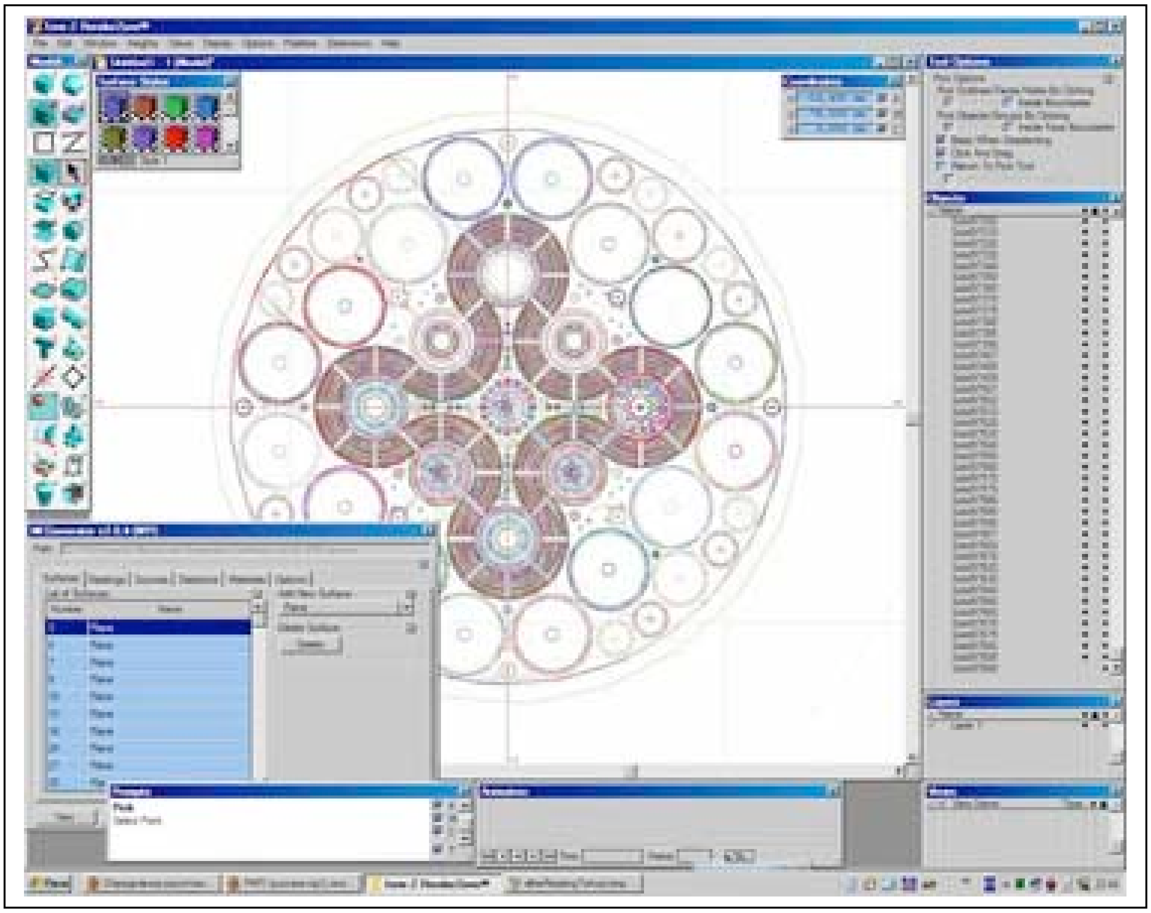


The example above shows the upper portion of the inner and outer containment vessels in the 9975 Type B shipping container. This COG model was translated from a Pro-E[10] model developed by the packaging designers at the Savannah River Site. This model makes extensive use of the REVOLUTION (closed surface) card in COG.

The semi-automatic translator codes developed to date are applicable to only a few specific types of problems and require considerable expertise by the user. To remedy this

[8] A surface-of-revolution defined by points.
[9] A topographical surface defined by points of elevation Z on an X-Y grid.
[10] Parametric Technology Corporation. http://www.ptc.com.

situation, an on-going collaboration between Lawrence Livermore National Laboratory (USA) with LLC "Strela" (Snezhinsk, Russia) is developing FormZ[11] based plug-ins[12] to facilitate the general geometry translation between COG, MCNP and TART codes. The FormZ representation of the Annular Test Reactor derived from an MCNP model published in the ICSBEP Handbook is shown below as an example.



It is hoped that continued development of FormZ-based plug-ins will eventually result in cost savings of countless man-hours and that this approach may be eventually extended to other codes such as KENO and MCU used in the ICSBEP.

Cross-section library databases

COG version 10 was distributed to RSICC with several cross-section libraries of interest to criticality safety practitioners. These include:

ENDFB6R7 the US National Nuclear Data Center (Brookhaven National Laboratory) Evaluated Nuclear Data File, Version 6, Release 7

---

[11] auto·des·sys, Inc. http://formz.com.
[12] http://www.strela.snz.ru/en/projects/b530321.html.

ENDL90 the 1990 version of the Lawrence Livermore National Laboratory's Evaluated Neutron Data Library

RED2002 the 2002 version of a hybrid library devised by Dr. Dermott E. Cullen of the Lawrence Livermore National Laboratory

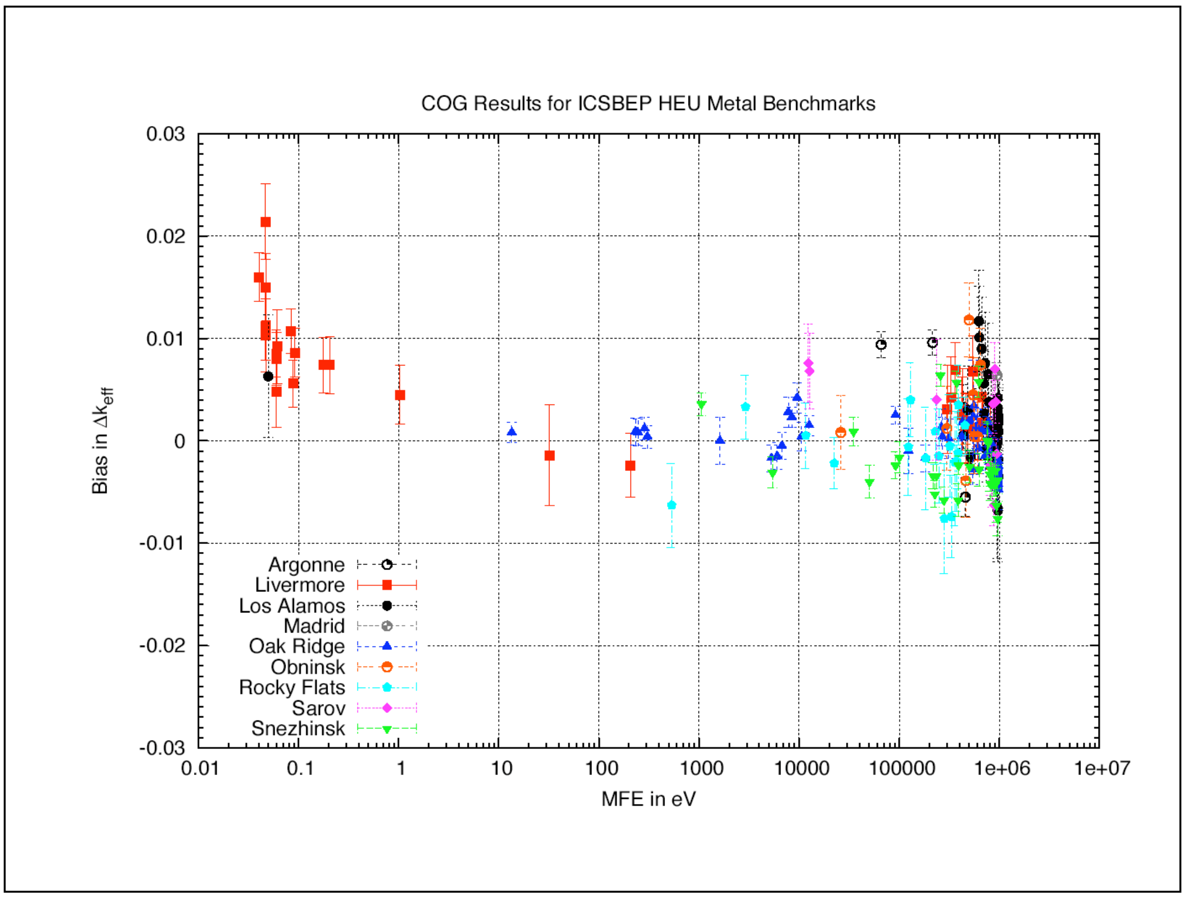
EPDL97 the 1997 version of the Lawrence Livermore National Laboratory Evaluated Photon Data Library

SAB3.0.296 the original 1968 General Atomics thermal scattering law "S(α,β)" data first released by the US NNDC as ENDF/B-III

SAB6.0.296 the ENDF/B-III data in ENDF-6 format first released by the US NNDC as ENDF/B-VI, Release 0

SAB6.2.296 the scattering law data generated at Los Alamos National Laboratory in 1993

COG also has electron, proton and photonuclear data that are not generally of interest to criticality safety practitioners.



Typical results of COG criticality calculations using ENDFB6R7 and SAB6.2.296 data are shown above for ICSBEP benchmarks for HEU metal. The 'bias' corresponds to the difference between COG calculated and ICSBEP benchmark-model k-eff values. MFE is the median energy of those neutrons causing fission events.

The results are generally excellent with the bias range within ±1% (Δk-eff) except for the Livermore cases moderated by beryllium oxide. It is hoped that changes to the beryllium cross-section to be released in ENDF/B-VII will improve these results.

Future activities

COG presently supports neutron data in ENDF-6, ENDL[13] and ACE formats. In the near future, the COG code development team plans to provide to RSICC an updated version with additional nuclear data libraries based on ENDF/B-VI (Release 8), ENDF/B-VII, JEFF and JENDL. LLNL validation efforts will focus primarily on ENDF/B-VII data for use in criticality safety applications. We plan to report these results on our website.

It is our hope that COG becomes another robust and independent computational method for general use by criticality safety practitioners.

---

[13] http://nuclear.llnl.gov/CNP/translation/